Handling DynamoDB Objects

Stefan Roman
DEVOPS ENGINEER

www.katapult.cloud

Understanding Basic Item Operations

AWS Command Line Tool

Configured AWS profile

Installed Python Packages

BOTO3

Appropriate privileges

All DynamoDB privileges

Understanding Basic Item Operations

Put Item – Creating Items

Get Item – Reading Items

Update Item – Updating Items

Delete Item – Deleting Items

Understanding Basic Item Operations

Batch Get Item

Read 100 items in one or more tables

Batch Write Item

Create or delete 25 items in tables

Understanding Basic Item Operations

```
{
    "Name": "Bob",
    "Age": 26,
    "Company": "Globomantics",
    "Position": "DevOps"
}
```

Understanding Basic Item Operations

```
{
   "Name": { "S": "Bob" },
   "Age": { "N": "26" },
   "Company": { "S": "Globomantics" },
   "Position": { "S": "DevOps" }
}
```

Understanding Basic Item Operations

```
"list": {
    "L": [
      {
        "S": "heart"
      },
      {
        "S": "lungs"
      }
    ]
  }
```

```
"map": {
    "M": {
      "age": {
        "N": "55"
      },
      "name": {
        "S": "Peter"
      }
    }
  }
```

Understanding Basic Item Operations

Put Item

ALL_OLD

Update Item

ALL_OLD

ALL_NEW

UPDATED_OLD

UPDATED_NEW

Delete Item

ALL_OLD

Demo

Utilize basic item operations against our *elaborate_employee_table* using:

- Console
- AWS CLI
- Python

Use return value function

Scanning Table Items

Scan

Reads and returns all table items

Filter Expressions

Filter results based on attributes or count

Consistency

Both strong and eventual consistency

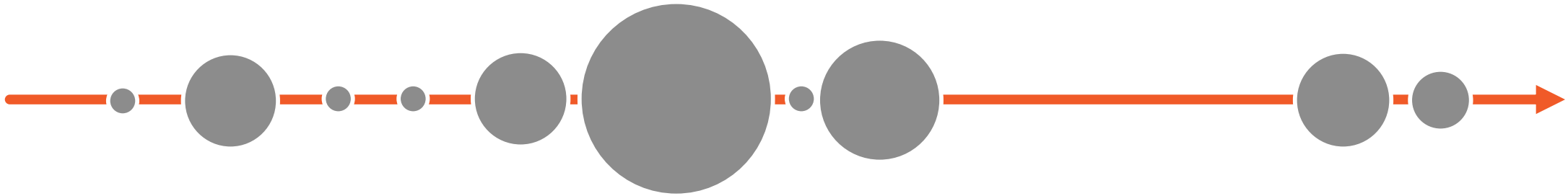
Scanning Table Items

Scan consumes huge amount of read capacity

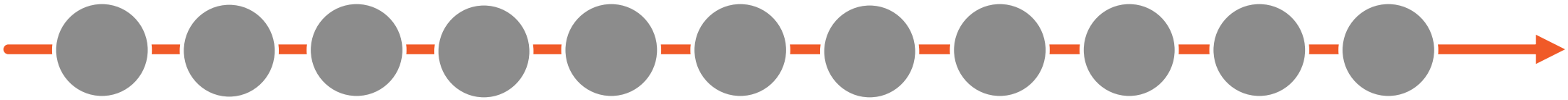
Pagination creates pauses between requests

Create copy of tables for scanning

Scanning Table Items



Scanning Table Items



Demo

Scan *elaborate_employee_table* using:

- Console
- AWS CLI
- Python

Use pagination function

Observe amount of consumed capacity

Querying Table Items

Query

Query items from base table and secondary index

Primary Key

Partition and sort keys can be used

Consistency

Both strong and eventual consistency

```
a = b
a < b
a <= b
a > b
a >= b
a BETWEEN b AND c
begins_with(attr, str)
```

Key Condition Expressions

- Equal
- Less than
- Less than or equal
- More than
- More than or equal
- Between
- Begins with

Querying Table Items

Expression Attribute Name

Avoid restricted attribute names

Expression Attribute Value

Avoid cluttered commands

Projection Expression

Avoid cluttered output from commands

Demo

Query *elaborate_employee_table* **using:**

- Console
- AWS CLI
- Python

Use:

- Key Condition Expression
- Expression Attribute Values
- Expression Attribute Names
- Projection Expression
- Filter Expression

Updating Table Items

```
{
  "id": 100,
  "prices": {
      "bed": 340,
      "pillow": 45,
      "blanket": 76,
      "mattress": 550
  }
}
```

Updating Table Items

```
get-item

id = 100

projection-expression

"prices.bed"
```

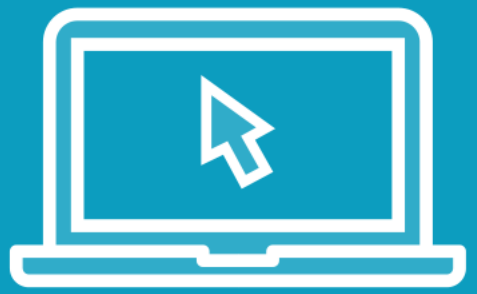
Updating Table Items

```
{
  "id": 100,
  "prices": [
    {
      "bedroom": {
        "bed": 340
      }
    }
  ]
}
```

Updating Table Items

```
get-item
id = 100
projection-expression
"prices[0].bedroom.bed"
```

Demo



Update items in *elaborate_employee_table* using:

- AWS CLI
- Python

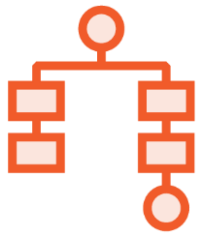
Use:

- Update expressions
- Nested attributes

Modifying Items Conditionally

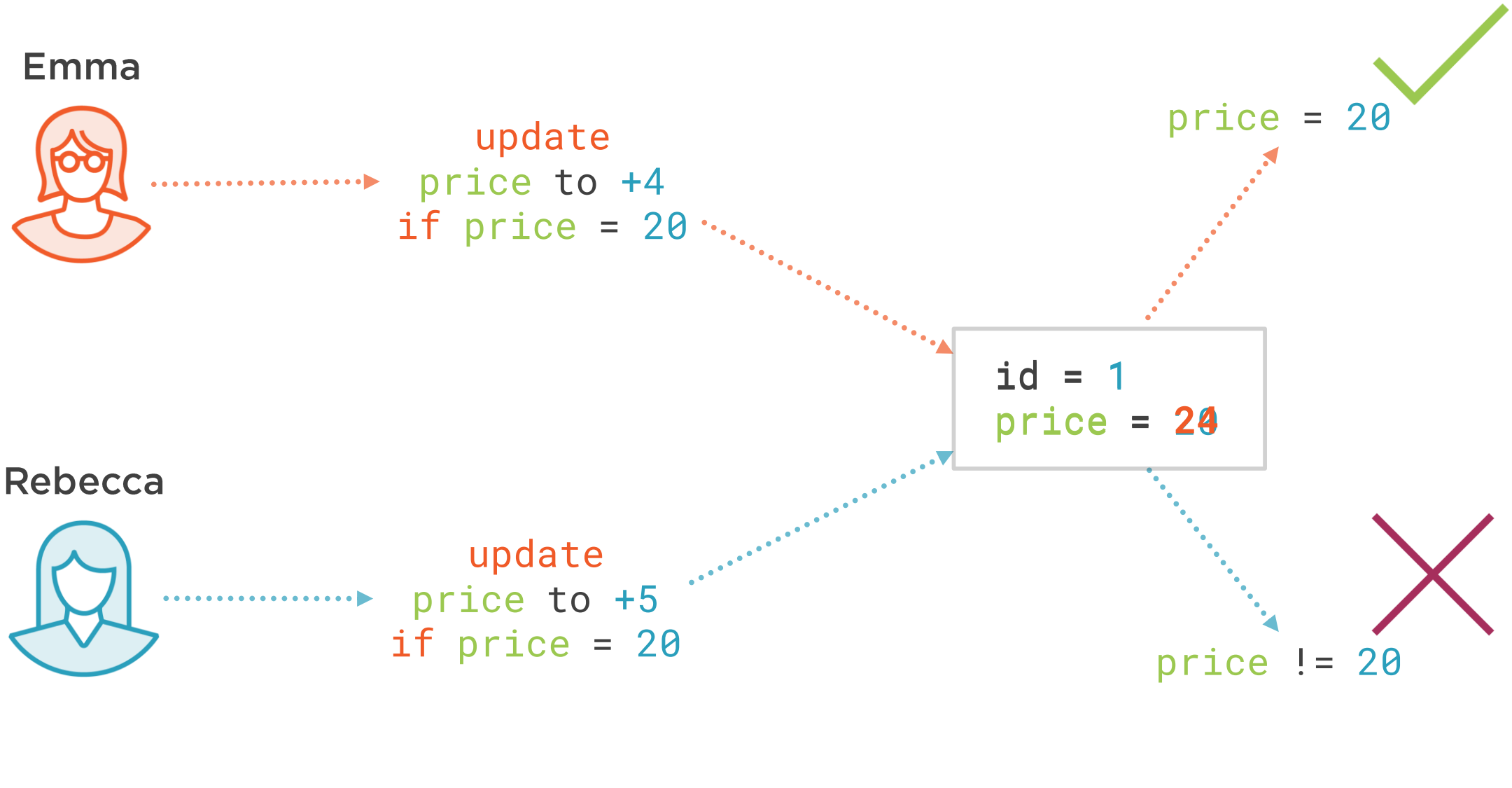
Write operations are all unconditional

Prevent accidental modification



Assures write idempotent operations

Modifying Items Conditionally



Modifying Items Conditionally

attribute_exists

attribute_not_exists

size

begins_with

attribute_type

contains

Demo

Use conditional writes on *elaborate_employee_table* using:

- AWS CLI
- Python

Explore and utilize conditional functions

Atomic Counters

Increments attribute value

Not idempotent

Used where accuracy does not matter

Demo

Update an item using atomic counters on *elaborate_employee_table* using:

- AWS CLI
- Python

Summary

Simple DynamoDB Operations

- Batch Operations
- AWS CLI
- Python

Table and Index Scanning

- Pagination Feature

DynamoDB Query Operations

- Expression Values
- Expression Names
- Projection Expression

Summary

Conditional and Unconditional Table Update Operations

- AWS CLI
- Python

Condition Expressions for Idempotency

Atomic Counters